

CS477 - Formal Software Development Methods

General Information and Introduction

Grigore Roşu

Department of Computer Science
University of Illinois at Urbana-Champaign

General Information

- Class Webpage and Newsgroup:
<http://fsl.cs.uiuc.edu/~grosu> (go to “Classes”)
<news://class.cs477>
- Lectures: Tuesdays/Thursdays 9:30 - 10:45, 1103 Siebel Center
- Office hours: Mondays 10:00 - 12:00, 2110 Siebel Center
- Instructor: **Grigore Roşu**
 - Office: 2110 Siebel Center
 - Email: grosu@cs.uiuc.edu
 - WWW: <http://cs.uiuc.edu/grosu>
 - Secretary: Andrea Whitesell
(2106 SC, whitesel@uiuc.edu, 333-1043)

Textbooks

No textbook required! Self contained lecture notes and links to URL's will be posted on class' webpage.

Grading

- Students registered for 4 units
 - Homework assignments (or MPs): **45%**
 - Final exam: **30%**
 - Individual project: **25%**
- Students registered for 3 units
 - Homework assignments: **60%**
 - Final exam: **40%**

The Homework Assignments

- The notions presented in class will be often backed by machine supported formalizations which you are supposed to modify or redo entirely as part of your assignments and as part of your project
- Assignments will be complete approximately every 4-6 lectures, on Tuesdays, and are due the next Tuesday.
- The exercises in the lecture notes are of two types: regular exercises and homework exercises:
 - *Exercise*. The regular exercises are intended to warm you up and to help you better understand other deeper notions.
 - *Homework Exercise*. The homework exercises are those which you have to solve as part of your homework.
- Therefore, you have *between 1 and 3-5 weeks for each exercise*.

The Unit Project

- The unit project will consist of using formal methods to develop, design or verify a software system. There is some degree of flexibility wrt the final project. If you want to do your own project based on your research interests, please let me know in advance, so we can together agree on what to do.

The Final Exam

- The final exam will test your overall understanding of the concepts discussed in class, and it is expected to be consistent with your homework assignments' scores.

Collaboration and Other Policies

- You are free to discuss the homework assignments with other students (and are encouraged to do so!). The focus of any such discussion should be limited to figuring the problem specification, not coming up with a solution. *You may not jointly write or code any assignment, unless explicitly otherwise stated.* To do so will be considered cheating! All cheating will be penalized by automatically assigning a failing grade for the course and instigating further disciplinary action with the appropriate university disciplinary body.
- You should retain copies of your assignments until you receive your final grade. In the event of a discrepancy between your scores on assignments and those on the exams, you may be asked to explain any work you performed. Your grade may be adversely affected by an inability to explain your work or by

failure to retain copies of it. All students are required to take the exam in order to receive a grade in the course.

- The course has a newsgroup. You are encouraged to use this group to ask questions, answer mundane system questions for other students, discuss homeworks, etc. In consideration for your peers, please don't use it to post flames, irrelevant messages, ads, etc.

Course Description

CS477/ECE478 is a course on Formal Methods and their use in software development.

Formal methods is a term that refers to a diverse collection of techniques, with strong mathematical foundations, that are used to provide assurance about the correctness of systems.

The course aims to familiarize students interested in software engineering with languages and methods for formal specification, development and verification. More specifically, the students will be exposed to the foundations and experimental use of the following formal methods tools:

- Maude, an executable specification language with tool support for formal analysis, including theorem proving and model checking;
- Java PathFinder and JavaFAN, two model checkers for Java; and
- JavaMOP, a runtime verification framework for Java.